

RANCANG BANGUN APLIKASI MASSIVE MULTIPLAYER ONLINE ROLE PLAYING GAME BERBASIS CLIENT-SERVER DENGAN OPTIMALISASI PAKET

Yogi Ario Kusuma¹, Hadi Chandra Kusuma¹, Isbat Uzzin Nadhori, S.Kom², Rizky Yuniar Hakkun²
Mahasiswa¹, Dosen²

Politeknik Elektronika Negeri Surabaya

Institut Teknologi Sepuluh Nopember

Kampus PENS-ITS Keputih Sukolilo Surabaya 60111

Telp (+62)31-5947280, 5946114, Fax. (+62)31-5946114

Email : yogiarka@gmail.com, hadi.chandra.kusuma@gmail.com, isbat@eepis-its.edu, rizky@eepis-its.edu

Abstrak

MMORPG (Massive Multiplayer Online Role Playing Game) adalah salah satu dari genre yang paling sukses dan populer. Masalah yang sering dihadapi oleh sebuah MMORPG adalah konsumsi bandwidth. Dengan banyak sekali pemain yang online dalam satu waktu, server diwajibkan untuk dapat melayani semua permintaan dari client, yang berakibat pada konsumsi bandwidth yang sangat besar. Cara mengatasinya adalah dengan optimalisasi. Optimalisasi dapat dilakukan dengan berbagai cara. Salah satu caranya adalah dengan mengoptimalkan paket-paket dengan mengubah tipe paket menjadi binary yang memiliki besar data yang kecil.

Pada proyek akhir ini, penyusun akan membangun sebuah MMORPG berbasis client server yang memiliki keunggulan dalam optimalisasi paket. Sehingga penggunaan jaringan dan bandwidth dapat lebih optimal dan semua request dari client dapat terpenuhi

Kata Kunci : MMORPG, Optimalisasi, Paket, Client, Server

1. Pendahuluan

1.1 Latar Belakang

MMORPG (Massive Multiplayer Online Role Playing Games) adalah salah satu dari genre yang paling sukses dan populer. Contohnya World of Warcraft, yang memiliki 11 juta pelanggan berbayar di seluruh dunia.

Masalah yang sering dihadapi oleh sebuah MMORPG adalah konsumsi bandwidth. Dengan banyak sekali pemain yang online dalam satu waktu, server diwajibkan untuk dapat melayani semua permintaan dari client, yang berakibat pada konsumsi bandwidth yang sangat besar. Salah satu caranya adalah dengan mengoptimalkan paket-paket yang dikirimkan ke client sehingga penggunaan jaringan dan bandwidth dapat lebih optimal dan semua request dari client dapat terpenuhi

Dalam proyek akhir ini akan dibuat MMORPG 2D dengan segala elemen dasar yang mempengaruhinya seperti AI (Artificial Intelligence), sprite, database, format paket, dan sebagainya. Juga akan dilakukan optimalisasi pada paket agar konsumsi bandwidth dapat digunakan dengan seoptimal mungkin.

1.2 Rumusan Permasalahan

Dalam perencanaan dan pembuatan perangkat lunak yang akan dibuat pada proyek akhir, terdapat beberapa permasalahan yang dihadapi, seperti :

1. Bagaimana cara server dan client berkomunikasi
2. Bagaimana cara server bisa membuat koneksi ke multiclient
3. Bagaimana server/client mengenali jenis dan kegunaan paket
4. Bagaimana cara mengoptimalkan paket pada komunikasi antara server dan client

1.3 Tujuan Proyek

Tujuan dari proyek akhir ini adalah membuat suatu perangkat lunak game dengan genre permainan Role Playing (RPG) yang dapat dimainkan oleh banyak orang secara bersama-sama dan online (Massive Multiplayer Online) berbasis client-server serta memiliki sistem paket yang teroptimalkan.

1.4 Kontribusi Proyek

Diharapkan dengan berhasilnya game ini, dapat mengatasi permasalahan yang sering terjadi pada game online serupa, yaitu masalah optimalisasi paket dan jaringan, dengan membuat konsumsi bandwidth seoptimal mungkin.

2. Teori Penunjang

2.1 Internet

Internet adalah sebuah sistem global dari jaringan komputer yang saling yang terinterkoneksi yang menggunakan standar *Internet Protocol Suite* (TCP/IP), yang melayani milyaran user di seluruh dunia. Internet memiliki jenis dan sumber informasi yang sangat bervariasi dan juga layanan-layanan, yang sebagian terbesar terdapat pada halaman *hypertext document* pada *World Wide Web* (WWW) dan infrastruktur untuk mendukung surat elektronik

2.2 LAN (Local Area Network)

LAN adalah jaringan komputer yang mencakup area fisik kecil, seperti rumah, kantor, atau kelompok kecil dari bangunan, seperti sekolah, atau bandara. Definisi karakteristik LAN, berbeda dengan jaringan luas (WAN), termasuk, kecepatan transfer data yang lebih besar, wilayah geografis yang lebih kecil, dan kurangnya kebutuhan untuk saluran telekomunikasi yang disewa.

2.3 MMORPG (Massively Multiplayer Online Role-Playing Game)

Massively multiplayer online role-playing game (MMORPG) adalah genre permainan komputer memainkan peran, di mana sejumlah besar pemain berinteraksi dengan satu sama lain dalam permainan di dalam dunia maya.

2.4 Client-Server

Client-server menggambarkan hubungan antara dua program komputer, di mana salah satu program, program client, membuat service request ke program lainnya, yaitu program server. Fungsi network standar seperti email, web access dan database access, bekerja berdasarkan prinsip client-server.

Prinsip client-server telah menjadi salah satu ide sentral dari network computing. Banyak sekali aplikasi bisnis menggunakan model ini. Sama halnya protokol aplikasi internet, seperti HTTP, SMTP, Telnet, DNS.

Setiap program client dapat mengirim data request ke satu atau lebih server yang terhubung. Sedangkan server dapat menerima (accept) request tersebut, memprosesnya dan mengembalikan informasi yang diminta kepada client. Konsep ini dapat diterapkan pada banyak aplikasi, tetapi tetap dengan dasar arsitektur yang sama.

2.5 TCP vs UDP

Pemilihan protokol mana yang akan digunakan dalam proses pembuatan online game adalah sangat penting. Ketika membuat sebuah turn-based online game di mana kecepatan tidak diutamakan, TCP adalah pilihan yang tepat karena lebih handal. Tetapi ketika kecepatan sangat diutamakan dan sangat sering terjadi proses update pada paket seperti FPS (First Person Shooting) game, maka UDP lebih cocok untuk menangannya.

a. TCP

TCP didesain agar paket yang dikirimkan pasti akan sampai kepada user yang dituju meskipun mengorbankan kecepatan pengiriman. Dengan TCP, kita tidak perlu mengkhawatirkan hilangnya sebuah paket yang dapat terjadi dengan UDP.

b. UDP

UDP didesain untuk mengirim paket ke user tujuan secara cepat. Paket dikirimkan dengan bentuk cluster, berbeda dengan TCP yang menggunakan sistem stream. Hal ini membuat paket-paket boleh datang dengan tidak berurutan.

Perbedaan antara keduanya dapat dilihat pada tabel di bawah ini :

TCP	UDP
Besar paket bervariasi / berubah-ubah	Besar paket tetap
Dijamin pasti sampai pada tempat tujuan	Tidak ada jaminan pasti sampai ke tempat tujuan
FIFO	Tidak perlu FIFO
Kecepatan lambat	Sangat cepat

Tabel 2.1 TCP vs UDP

2.6 Socket

Socket adalah interface pada jaringan yang menjadi titik komunikasi antarmesin pada Internet Protocol, dan tentunya tanpa komunikasi ini, tidak akan ada pertukaran data dan informasi jaringan.

Socket address adalah kombinasi dari alamat IP dan sebuah port (yang berhubungan dengan proses suatu aplikasi) yang menjadi satu identitas. Sebuah internet socket memiliki ciri-ciri khusus yang dibentuk oleh kombinasi unik dari :

- Protocol (TCP,UDP)
- Alamat socket lokal (Local Socket Address) yang terdiri dari Local IP address dan nomor port
- Remote socket address (Hanya untuk TCP socket)

2.7 Winsock

Winsock adalah singkatan dari Windows Socket, yaitu sebuah API yang mengijinkan aplikasi Windows-based untuk mengakses transport protokol. Aplikasi yang dapat menulis ke Winsock interface termasuk File Transfer Protocol (FTP) dan Simple Network Management Protocol (SNMP).

2.8 Paket

Paket adalah kesatuan data terformat yang dibawa oleh jaringan komputer. Komunikasi komputer tradisional yang tidak mendukung paket, seperti point-to-point, mengirimkan data sebagai serangkaian byte, karakter, atau bit. Ketika data diformat menjadi paket-paket, bitrate medium komunikasi dapat dibagi dengan lebih baik di antara pengguna daripada jika jaringan yang circuit switched. Dengan menggunakan jaringan packet switched sulit untuk menjamin kecepatan bit serendah mungkin.

2.9 Struktur Paket

Sebuah paket terdiri atas dua bagian yang paling penting, yaitu header dan data.

- **Header**

Header adalah single byte yang menyatakan bagaimana cara menangani data. Contoh sebuah header :

DataCode.User_Move

Ketika header di atas dikirim oleh client kepada server, maka server akan mengetahui bahwa seluruh data pada merupakan informasi bagaimana cara menggerakkan user atau karakter di dalam game.

- **Data**

Isi dan kegunaan data di dalam paket tergantung dari informasi header yang dimilikinya. Setiap DataCode yang valid ditangani oleh sebuah sub yang memiliki nama yang sama dan diawali dengan **Data_**. Sehingga header di atas akan memiliki sub seperti ini :

Sub Data_User_Move

2.10 Jenis Paket

- **String Packets**

Struktur paket yang kebanyakan digunakan pada ORPG adalah berupa string packets. String packet adalah sebuah paket yang dikirimkan dengan bentuk string. String sendiri terdiri atas kumpulan karakter-karakter ASCII, yang memiliki nilai antara 0-255. Sedangkan paket adalah sebagian dari informasi yang dikirim dari satu komputer ke komputer yang lain melalui jaringan atau internet.

- **Binary Packets**

Struktur paket terdiri atas :

- **DataID**
Adalah tanda pengenal pada paket, paket tersebut akan digunakan untuk apa. Semisal untuk pergerakan karakter, attacking dan sebagainya.
- **Data**
Adalah data aktual pada paket yang dapat berisi apa saja yang berhubungan dengan DataID
- **Separator**
Seringkali digunakan untuk memisahkan antara data pada paket untuk variable-length data, jadi dapat diketahui ketika satu bagian data berakhir dan bagian lainnya mulai.
- **EndID**
Seringkali digunakan dalam bentuk single karakter yang menyatakan berakhirnya suatu paket.

Pada umumnya struktur paket adalah sebagai berikut

<DataID>(<Data><Separator><Data><Separator>...)<EndID>

Kelemahan struktur di atas adalah DataID biasanya terdiri atas banyak karakter yang panjang, kemudian dilanjutkan dengan Separator yang digunakan untuk memisahkan data string, dan akhirnya EndID yang menyatakan berakhirnya paket tersebut. Ini terlihat tidak

efektif ketika dibandingkan dengan struktur lain seperti ini

<DataID(1 Byte)>(<Data><Data>...)

Struktur di atas terlihat sangat simpel, tanpa Separator dan EndID, hanya DataID dan Data. Inilah alasannya mengapa paket ditulis dalam bentuk binary, lebih mudah dan simpel serta tidak menghabiskan banyak bandwidth ketika dikirim.

- **Byte Array**

Paket yang digunakan pada Proyek Akhir ini ditulis dalam bentuk byte array. Mengapa? Alasannya karena byte array paling mudah diorganisir, semua tipe data cocok di dalamnya dan beberapa socket dapat dioptimalisasi untuk mengirimkan byte dalam bentuk array.

2.11 Data Buffer

Buffer merupakan suatu area di dalam memori yang digunakan untuk menyimpan data sementara ketika sedang dipindahkan dari satu tempat ke tempat lain. Biasanya, data disimpan dalam buffer karena diambil dari sebuah perangkat input (seperti keyboard) atau sebelum dikirim ke perangkat output (seperti printer). Namun, buffer dapat digunakan bila memindahkan data antara proses-proses di dalam komputer. Hal ini sebanding dengan buffer di bidang telekomunikasi. Buffer dapat diimplementasikan baik dalam perangkat keras atau perangkat lunak, namun sebagian besar buffer yang diimplementasikan dalam perangkat lunak. Buffer biasanya digunakan ketika terdapat perbedaan antara tingkat di mana data yang diterima dan tingkat di mana ia dapat diproses, atau dalam hal bahwa angka ini adalah variabel, misalnya dalam sebuah spooler printer atau dalam video online streaming.

2.12 Bandwidth

Dalam jaringan komputer dan ilmu komputer, bandwidth, bandwidth jaringan, data bandwidth atau bandwidth digital, adalah ukuran bitrate dari sumber daya komunikasi data dan ditampilkan dalam bit/detik atau kelipatan dari itu (kilobit/detik, megabits/detik dll).

Bandwidth dalam bit/ s kadang-kadang dapat berarti kecepatan bit bersih (juga dikenal sebagai bit rate puncak), kapasitas saluran, atau *throughput* maksimum jalur komunikasi logikal atau fisik dalam sistem komunikasi digital. Misalnya, tes bandwidth mengukur throughput maksimal dari sebuah jaringan komputer.

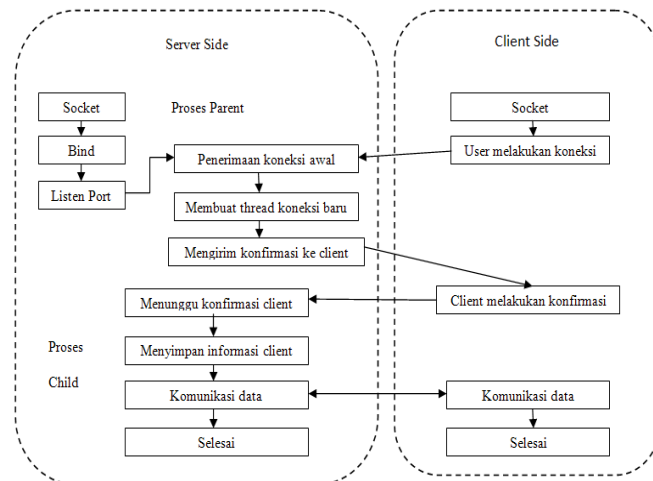
2.13 Wireshark

Wireshark adalah program gratis dan open source yang digunakan untuk menganalisa paket. Dapat dipakai untuk mengatasi masalah jaringan, analisis, dan pengembangan perangkat lunak protokol komunikasi, dan pendidikan. Awalnya bernama Ethereal, tetapi pada Mei 2006 proyek ini dinamai Wireshark karena masalah merek dagang.

Wireshark merupakan program *cross-platform*, menggunakan toolkit GTK widget + untuk mengimplementasikan antarmuka pengguna, dan

menggunakan pcap untuk menangkap paket, dan berjalan pada berbagai sistem operasi mirip Unix termasuk Linux, Mac OS X, BSD, dan Solaris, dan Microsoft Windows. Ada juga versi berbasis terminal (non-GUI) untuk Linux yang disebut TShark. Wireshark, dan program lain yang didistribusikan dengan itu seperti TShark, adalah perangkat lunak bebas, dirilis di bawah GNU General Public License.

3. Rancangan dan Implementasi Sistem



Gambar 3.1 Rancangan Sistem

Secara garis besar, blok diagram di atas merupakan acuan untuk perancangan sistem. Setiap kali client baru akan melakukan koneksi dengan server, maka server akan melakukan proses dari “Penerimaan koneksi awal” sampai dengan “Selesai”. Maka dari itu server dapat melakukan koneksi dengan lebih dari satu client.

3.1 JARINGAN

Pada sub-bab ini akan dijelaskan segala sesuatu yang berhubungan dengan perancangan dan pembuatan jaringan dan apa saja yang dibutuhkan dalam jaringan tersebut

3.1.1 KONEKSI

Koneksi pada sistem menggunakan socket, dengan tipe koneksi adalah TCP/IP. Sistem tidak menggunakan UDP, karena ketika membuat sebuah MMORPG di mana kecepatan tidak diutamakan, TCP adalah pilihan yang tepat karena lebih handal. Untuk UDP sendiri akan lebih handal ketika kecepatan sangat diutamakan dan sangat sering terjadi proses update pada paket seperti FPS (First Person Shooting) game.

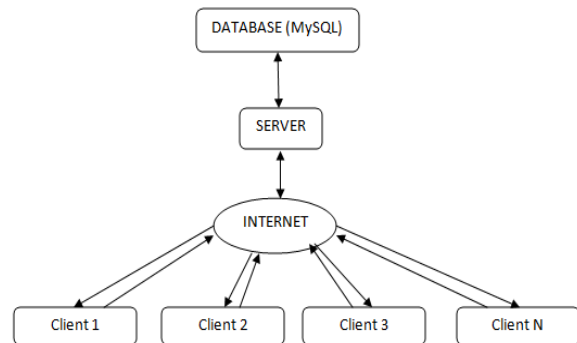
3.1.2 KOMUNIKASI DATA

Untuk melakukan komunikasi data antara server dan client, data dikirimkan dalam bentuk DataID, yang terdapat pada kedua sisi server dan

client. DataID memiliki format <DataID><Parameter>.

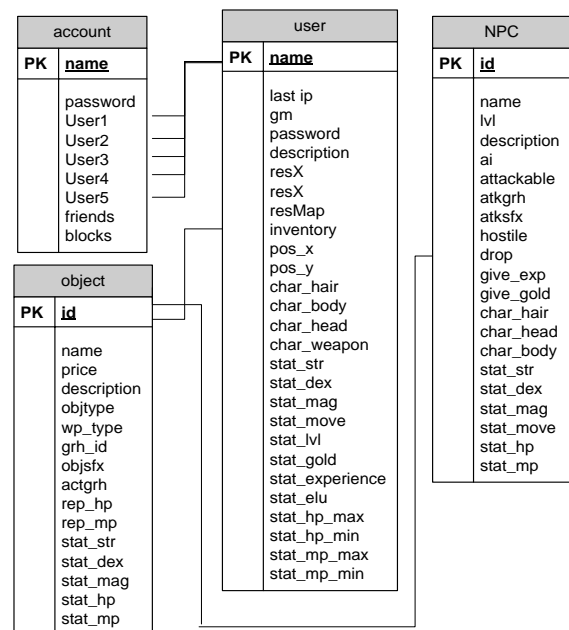
3.2 DATABASE

Database yang digunakan pada sistem ini adalah MySQL. Struktur hubungan database ke jaringan pada koneksi antara client dan server dapat dilihat pada gambar/bagan berikut ini :



Gambar 3.2 Struktur hubungan database ke jaringan

Berikut merupakan diagram struktur dan relationship dari tabel yang digunakan dalam database sistem :



Gambar 3.3 Struktur tabel dan relationship

Struktur di atas merupakan struktur database dari sistem, di mana sistem memiliki empat tabel utama, yaitu *account*, *user*, *object*, dan *NPC*. Sedangkan PK merupakan primary key dari setiap tabel, yang berhubungan dan menjadi foreign key pada tabel lain.

4. Hasil Dan Kesimpulan

Berdasarkan analisa dari beberapa pengujian yang diterangkan pada bab sebelumnya, kesimpulan yang didapatkan adalah

1. Semakin banyak client yang terhubung pada server, maka rata-rata bytes/detik yang dikirimkan akan semakin besar, begitu juga dengan rata-rata paket/detik, sesuai dengan hasil Uji Coba I pada Bab IV.
2. Sistem (Pensaga) sejauh ini dapat berjalan di 28 client, sesuai dengan hasil Uji Coba II pada Bab IV.
3. Pengiriman paket dengan menggunakan optimalisasi paket akan menggunakan bandwidth yang jauh lebih sedikit dibandingkan dengan tidak menggunakan optimalisasi paket. Terbukti dari perbandingan antar Pensaga yang teroptimalisasi dengan yang tidak teroptimalisasi pada hasil Uji Coba III di Bab IV.

Daftar Pustaka

- [1]. Client-Server, <http://en.wikipedia.org/wiki/Client-Server>.
[2]. Winsock, <http://en.wikipedia.org/wiki/Winsock>. Barron, [3]. Todd. *Multiplayer Game Programming*. Prima Publishing, 2001.
[4]. Sánchez, Daniel. *Core Techniques and Algorithms in Game Programming*. New Riders Publishing, 2003.

[CV Penulis]

Hadi Chandra Kusuma, menjalankan studi D4 bidang Teknik Informatika pada Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember (PENS-ITS) semester 8.

Yogi Ario Kusuma, menjalankan studi D4 bidang Teknik Informatika pada Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember (PENS-ITS) semester 8.